

WHAT IS CLAIMED IS:

1. A method for applying at least one receiving attribute at a receiving destination, comprising:

5 specifying receiving attributes based on constraints on definable multivariate functions.

2. The method of claim 1, wherein said definable multivariate functions comprise linear combinations of products of univariate functions of attribute values.

3. The method of claim 1, wherein said definable multivariate functions comprise stack functions.

10 4. The method of claim 2, wherein the receiving attributes specified comprise limit constraints applicable to functions of said receiving attributes.

5. The method of claim 2, wherein the receiving attributes specified comprise optimization constraints applicable to functions of said receiving attributes.

15 6. The method of claim 2, wherein the products comprise product terms, and wherein definable multivariate functions comprise at least one from a group comprising:

number of product terms in the linear combination;

number of elements in each product term;

attribute codes for attributes in each product term;

20 function codes for univariate functions on attribute values; and

multipliers for at least one linear combination.

7. The method of claim 4, wherein the limit constraints comprise at least one from a group comprising:

maximum supportable values for the receiving destination for at least one constraint on the definable multivariate functions; and

minimum supportable values for the receiving destination for at least one constraint on the definable multivariate functions.

5 8. The method of claim 5, wherein optimization constraints comprise specify at least one from a group comprising:

minimization of attribute measures; and

maximization of attribute measures.

10 9. The method of claim 3, wherein the receiving attributes specified comprise limit constraints applicable to functions of said receiving attributes.

10 10. The method of claim 3, wherein the receiving attributes specified comprise optimization constraints applicable to functions of said receiving attributes.

11. A method for enabling comparison between scaling attributes and receiving attributes at a receiving destination, comprising:

15 specifying receiving attributes based on measures of the attributes; and imposing constraints on the measures of the attributes.

12. The method of claim 9, wherein measures of the attributes comprise a linear combination of products of univariate functions of attribute values.

20 13. The method of claim 11, wherein measures of the attributes further comprise at least one constraint on definable multivariate functions.

14. The method of claim 11, wherein the receiving attributes specified comprise limit constraints applicable to functions of said receiving attributes.

15. The method of claim 11, wherein the receiving attributes specified comprise optimization constraints applicable to functions of said receiving attributes.

16. The method of claim 13, wherein the products comprise product terms, and wherein definable multivariate functions comprise at least one from a group comprising:

number of product terms in the linear combination;

5 number of elements in each product term;

attribute codes for attributes in each product term;

function codes for univariate functions on attribute values; and

multipliers for at least one linear combination.

17. The method of claim 14, wherein the limit constraints comprise at least
10 one from a group comprising:

maximum supportable values for the receiving destination for at least one measure; and

minimum supportable values for the receiving destination for at least one measure.

18. The method of claim 15, wherein optimization constraints comprise
15 specify at least one from a group comprising:

minimization of attribute measures; and

maximization of attribute measures.

19. The method of claim 11, wherein measures of the attributes comprise
20 stack functions.

20. A method of processing functions at a format independent adaptation engine in a network using XML syntax, the method comprising:

specifying an expression for the function using an ordered list of numeric constants, variables, arguments, and operators pushed into an expression stack to evaluate the expression;

5 wherein said functions comprise at least one from a group comprising constraints applied by the adaptation engine, sequence field operations, and offset length calculations.

21. The method of claim 20, further comprising evaluating the expression for the function after said specifying.

10 22. The method of claim 21, wherein evaluating the expression for the function comprises serializing the desired expression into a serialized expression and applying elements of the serialized expression to a stack in order.

23. The method of claim 22, wherein applying elements of the serialized expression to the stack in order comprises applying numeric constants to the stack and applying any operators to at least one topmost stack element.

15 24. The method of claim 23, wherein applying elements of the serialized expression further comprise setting variables to appropriate numeric constants and applying the numeric constants to the stack.

20 25. The method of claim 23, wherein unary operators operate on a topmost numeric stack element and binary operators apply operate on two topmost numeric stack elements.

26. A method of processing at least one predetermined function at a format independent adaptation engine, the method comprising:

25 specifying an expression for the function using an ordered list of numeric constants, variables, arguments, and operators pushed into an expression stack to evaluate the expression;

wherein said method employs a markup language.

27. The method of claim 26, further comprising evaluating the expression for the function after said specifying.

28. The method of claim 27, wherein evaluating the expression for the function comprises serializing the desired expression into a serialized expression and
5 applying elements of the serialized expression to a stack in order.

29. The method of claim 28, wherein applying elements of the serialized expression to the stack in order comprises applying numeric constants to the stack and applying any operators to at least one topmost stack element.

30. The method of claim 29, wherein applying elements of the serialized
10 expression further comprise setting variables to appropriate numeric constants and applying the numeric constants to the stack.

31. The method of claim 29, wherein unary operators operate on a topmost numeric stack element and binary operators apply operate on two topmost numeric stack elements.

15 32. The method of claim 26, wherein the predetermined function comprises at least one from a group comprising a constraint function, a sequence field function, and an offset function.

33. A transcoder for comparing between scaling attributes and receiving attributes, comprising:

20 a specification apparatus for specifying receiving attributes based on measures of the attributes; and

a constraint imposing apparatus for imposing constraints on the measures of the attributes.

34. A system for applying receiving attributes, comprising:

25 a transcoder for comparing between scaling attributes and receiving attributes, comprising:

a specification apparatus for specifying receiving attributes based on measures of the attributes; and

a constraint imposing apparatus for imposing constraints on the measures of the attributes.

- 5 35. A transcoder for processing at least one predetermined function, comprising:

 a function expression specifier for specifying an expression for the function using an ordered list of numeric constants, variables, arguments, and operators pushed into an expression stack to evaluate the expression;

- 10 wherein said transcoder employs a markup language.

36. A system for processing functions, comprising:

 a transcoder for processing at least one predetermined function, comprising:

- a function expression specifier for specifying an expression for the function using an ordered list of numeric constants, variables, arguments, and
15 operators pushed into an expression stack to evaluate the expression;

 wherein said transcoder employs a markup language.